AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Canceled)

2. (Currently Amended) A network processor, comprising:

a fetch control unit, having an input coupled to receive an execution feedback signal with stall information related to a plurality of threads on a per thread basis, the fetch control unit generating an instruction fetch sequence based on the execution feedback signal;

an instruction cache, having an input coupled to an output of the fetch control unit, the instruction cache dispatching instruction data responsive to the instruction fetch sequence; and

~~The network processor of claim 1, further comprising~~ an instruction queue having an output coupled to the fetch control unit input, the instruction queue generating a queue feedback signal responsive to a thread queue condition associated with a thread from the plurality of threads, wherein the fetch control unit generates the instruction fetch sequence also based on the queue feedback signal.

3. (Original) The network processor of claim 2, wherein the thread queue condition indicates that a thread queue has less than a first amount of remaining storage.

4. (Original) The network processor of claim 3, wherein the fetch control unit blocks the thread from the instruction fetch sequence responsive to the queue feedback signal.

5. (Original) The network processor of claim 2, wherein the thread queue condition indicates that a thread queue has less than a second amount of remaining decoded instructions.

6. (Original) The network processor of claim 5, wherein the fetch control unit advances the thread in the instruction fetch sequence responsive to the queue feedback signal.

7. (Currently Amended) The network processor of claim ~~1~~ 2, further comprising a thread interleaver having an output coupled to the fetch control unit input, the thread interleaver generating an interleaver feedback signal responsive to a thread condition, wherein the fetch control unit generates the instruction fetch sequence also based on the interleaver feedback signal.

8. (Original) The network processor of claim 7, wherein the thread condition indicates that a thread from the plurality of threads is ineligible for execution.

9. (Currently Amended) A network processor, comprising:

a fetch control unit, having an input coupled to receive an execution feedback signal with stall information related to a plurality of threads on a per thread basis, the fetch control unit generating an instruction fetch sequence based on the execution feedback signal;

an instruction cache, having an input coupled to an output of the fetch control unit, the instruction cache dispatching instruction data responsive to the instruction fetch sequence; and

a thread interleaver having an output coupled to the fetch control unit input, the thread interleaver generating an interleaver feedback signal responsive to a thread condition, wherein the fetch control unit generates the instruction fetch sequence also based on the interleaver feedback signal, and ~~The network processor of claim 7,~~ wherein the thread interleaver generates a thread execution sequence independent of the instruction fetch sequence.

10. (Currently Amended) The network processor of claim ~~1~~ 2, further comprising an execution pipeline having an output coupled to the fetch control unit input, the execution pipeline generating the execution feedback signal responsive to an execution stall.

11. (Original) The network processor of claim 10, wherein the fetch control unit delays the thread in the instruction fetch sequence responsive to the execution stall.

12. (Original) The network processor of claim 10, wherein the execution stall comprises one from the group consisting of a misprediction, an exception, a data cache miss, an external resource stall, an interlock, and a memory operation ordering.

13. (Currently Amended) The network processor of claim ~~1~~ 2, wherein the fetch control unit generates the instruction fetch sequence, in a default state, by selecting a thread from the plurality of threads according to round robin arbitration.

14. (Currently Amended) The network processor of claim ~~1~~ 2, wherein the execution feedback signal is capable of including stall information related to each of the plurality of threads.

15. (Currently Amended) The network processor of claim ~~1~~ 2, wherein the instruction data are packet processing instructions related to at least one form the group consisting of: packet routing, switching, bridging and forwarding.

16. (Canceled)

17. (Currently Amended) A network processor, comprising:

means for fetching, having an input coupled to receive an execution feedback signal with stall information related to a plurality of threads on a per thread basis, the means for fetching generating an instruction fetch sequence based on the execution feedback signal;

means for storing, having an input coupled to an output of the means for fetching, the means for storing dispatching instruction data responsive to the instruction fetch sequence; and

~~The network processor of claim 16, further comprising~~ a means for queuing having an output coupled to the means for fetching input, the means for queuing generating a queue feedback signal responsive to a thread queue condition associated with a thread from the plurality of threads, wherein the means for fetching generates the instruction fetch sequence also based on the queue feedback signal.

18. (Original) The network processor of claim 17, wherein the thread queue condition indicates that a means for thread queuing has less than a first amount of remaining storage.

19. (Original) The network processor of claim 18, wherein the means for fetching blocks the thread from the instruction fetch sequence responsive to the queue feedback signal.

20. (Original) The network processor of claim 17, wherein the thread queue condition indicates that a means for thread queuing has less than a second amount of remaining decoded instructions.

21. (Original) The network processor of claim 20, wherein the means for fetching advances the thread in the instruction fetch sequence responsive to the queue feedback signal.

22. (Currently Amended) The network processor of claim ~~16~~ 17, further comprising a means for interleaving having an output coupled to the means for fetching input, the means for interleaving generating an interleaver feedback signal responsive to a thread condition, wherein the means for fetching generates the instruction fetch sequence also based on the interleaver feedback signal.

23. (Original) The network processor of claim 22, wherein the thread condition indicates that a thread from the plurality of threads is ineligible for execution.

24. (Currently Amended) A network processor, comprising:

means for fetching, having an input coupled to receive an execution feedback signal with stall information related to a plurality of threads on a per thread basis, the means for fetching generating an instruction fetch sequence based on the execution feedback signal;

means for storing, having an input coupled to an output of the means for fetching, the means for storing dispatching instruction data responsive to the instruction fetch sequence; and

a means for interleaving having an output coupled to the means for fetching input, the means for interleaving generating an interleaver feedback signal responsive to a thread condition, wherein the means for fetching generates the instruction fetch sequence also based on the interleaver feedback signal, and ~~The network processor of claim 22,~~ wherein the means for interleaving generates a thread execution sequence independent of the instruction fetch sequence.

25. (Currently Amended) The network processor of claim ~~16~~ 17, further comprising an means for executing having an output coupled to the means for fetching input, the

means for executing generating the execution feedback signal responsive to an execution stall.

26. (Original) The network processor of claim 25, wherein the means for fetching delays the thread in the instruction fetch sequence responsive to the execution stall.

27. (Original) The network processor of claim 25, wherein the execution stall comprises one from the group consisting of a data cache miss, an external resource stall, an interlock, and a memory operation ordering.

28. (Currently Amended) The network processor of claim ~~16~~ 17, wherein the means for fetching generates the instruction fetch sequence, in a default state, by selecting a thread from the plurality of threads according to round robin arbitration.

29. (Currently Amended) The network processor of claim ~~16~~ 17, wherein the execution feedback signal is capable of including stall information related to each of the plurality of threads.

30. (Currently Amended) The network processor of claim ~~16~~ 17, wherein the instruction data are packet processing instructions related to at least one form the group consisting of: packet routing, switching, bridging and forwarding.

31. (Canceled)

32. (Canceled)

33. (Canceled)

34. (Currently Amended) A method for fetching instructions in a network processor, comprising:

generating a queue feedback signal responsive to a thread queue condition associated with a thread from the plurality of threads, wherein the thread queue condition indicates that a thread queue has less than a first amount of remaining storage;

generating an instruction fetch sequence based on an execution feedback signal with stall information related to a plurality of threads on a per thread basis, comprising:

~~The method of claim 33, wherein the~~ generating the instruction fetch sequence ~~comprises~~ blocking the thread from the instruction fetch sequence responsive to the queue feedback signal; and

dispatching instruction data responsive to the instruction fetch sequence.

35. (Currently Amended) The method of claim ~~32~~ 34, wherein the thread queue condition indicates that a thread queue has less than a second amount of remaining decoded instructions.

36. (Currently Amended) A method for fetching instructions in a network processor, comprising:

generating a queue feedback signal responsive to a thread queue condition associated with a thread from the plurality of threads, wherein the thread queue condition indicates that a thread queue has less than a second amount of remaining decoded instructions;

generating an instruction fetch sequence based on an execution feedback signal with stall information related to a plurality of threads on a per thread basis, comprising:

~~The method of claim 35, wherein the generating the instruction fetch sequence comprises~~ advancing the thread in the instruction fetch sequence responsive to the queue feedback signal;

dispatching instruction data responsive to the instruction fetch sequence.

37. (Currently Amended) The method of claim ~~31~~ 36, wherein the generating the instruction fetch sequence further comprises generating an interleaver feedback signal responsive to a thread condition in a thread interleaver.

38. (Original) The method of claim 37, wherein the thread condition indicates that a thread from the plurality of threads is ineligible for execution.

39. (Currently Amended) A method for fetching instructions in a network processor, comprising:

generating an instruction fetch sequence based on an execution feedback signal with stall information related to a plurality of threads on a per thread basis, comprising:

generating an interleaver feedback signal responsive to a thread condition in a thread interleaver, comprising:

~~The method of claim 37, wherein the generating the feedback signal comprises~~ generating a thread execution sequence independent of the instruction fetch sequence; and

dispatching instruction data responsive to the instruction fetch sequence.

40. (Currently Amended) The method of claim ~~31~~ 39, the generating the instruction fetch sequence further comprises generating the execution feedback signal responsive to an execution stall.

41. (Original) The method of claim 40, wherein the generating the instruction fetch sequence further comprises delaying the thread in the instruction fetch sequence responsive to the execution stall.

42. (Original) The method of claim 40, wherein the execution stall comprises one from the group consisting of a data cache miss, an external resource stall, an interlock, and a memory operation ordering.

43. (Currently Amended) The method of claim ~~31~~ 39, wherein the generating the instruction fetch sequence comprises generating the instruction fetch sequence, in a default state, by selecting a thread from the plurality of threads according to round robin arbitration.

44. (Currently Amended) The method of claim ~~31~~ 39, wherein the feedback signal is capable of including stall information related to each of the plurality of threads.

45. (Currently Amended) The method of claim ~~31~~ 39, wherein the instruction data are packet processing instructions related to at least one form the group consisting of: packet routing, switching, bridging and forwarding.

46. (Canceled)

47. (Canceled)